Document Number: GRIS-SAM
Revision: -
Date: 28 January 1997

TECHNICAL REPORT
SYSTEM ADMINISTRATORS MANUAL (SAM)
FOR
GLOBAL RECONNAISSANCE INFORMATION SYSTEM (GRIS)

CONTRACT NO. N00039-96-C-00029
CDRL SEQUENCE NO. A004

Sponsor:

Ms. Anita Washington
Space and Naval Warfare Systems Command
2451 Crystal Drive, CPK5
Arlington, VA 22245-5200

Prepared By:

PRC, Inc.
1500 PRC Drive
McLean, VA 22102

Table of Contents

1.0 Introduction

1.1 Identification..

This System Administrator's Manual (SAM) provides system administrator personnel information on the installation and configuration of the Global Reconnaissance Information System (GRIS) Computer Software Configuration Item (CSCI), documenting the full range of configurable items for GRIS execution.

1.2 System Overview

The Global Reconnaissance Information System (GRIS) is a Joint Mission Application Software (JMAS) segment, within the Global Command and Control System (GCCS) domain for the Department of Defense. Under the program sponsorship of the Defense Information Systems Agency (DISA), Center For Applications Engineering, Command and Control Applications Department (JEKE) it is used by the Joint Reconnaissance Centers (JRCs) at designated Unified Commander-In-Chief (CINC) sites. GRIS provides automated support in planning, scheduling, reporting, and monitoring reconnaissance activities under the Sensitive Reconnaissance Operations (SRO) program in accordance with Enclosure (C) (Reconnaissance Reporting Instructions) of Chairman Joint Chiefs of Staff Instruction (CJCSI) 3250.01 (Series) - Policy Guidance For Sensitive Airborne and Maritime Surface Reconnaissance Operations.

GRIS maintains a near real-time status of all SRO missions and provides immediate on-line retrieval of mission, track, and message data. To accomplish this, GRIS provides automated real-time capture and processing of Reconnaissance Information Processing System (RIPS) format messages, and maintains a mission and track database containing schedule and resultant information. GRIS generates and releases outgoing SRO messages to the Automated Digital Network (AUTODIN) and provides on-line query and report capabilities detailing message, mission status, and scheduling information. It is used to maintain current Track Dictionary data and to generate the master copy of each new dictionary or set of change pages. GRIS has an external interface with the GCCS Automated Message Handling System (AMHS).

GRIS is installed at the National Military Command Center (NMCC) for the Joint Staff, U.S. Atlantic Command (ACOM), U.S. Central Command (CENTCOM), U.S. European Command (EUCOM), U.S. Pacific Command (PACOM), and at the U.S. Southern Command (SOUTHCOM). Each installation is operated independently, with information passed between GRIS sites and other participating organizations using RECON 1/2/3/4 format messages via AUTODIN.

1.3 Document Overview

Section 1 provides a system identification and overview, and an overview of this document. Sections 2 and 3 contain a summary and detailed description, respectively, of GRIS installation and configuration values including details on each of GRIS' configurable files. Section 4 provides a listing of script files used for GRIS. Section 5 provides a list of acronyms.

2.0 Installation And Configuration Summary

This section describes the typical minimum changes that each site will want to make to the configurable GRIS files. These are suggestions based on past installations.

2.1 Installation Instructions

This application is intended for use by the Joint Recon Commities at these sites: PACOM, EUCOM, ACOM, SOUTHCOM, CENTCOM AND NMCC.

If you are not one of these sites, DO NOT INSTALL the GRIS SEGMENT.

If you are a GRIS site:

1. Contact the GRIS user before installing the segment.

2. Determine where it makes sense to install GRIS.

3. Only install GRIS where it is NEEDED. DO NOT INSTALL GRIS ON EVERY CLIENT! Typically, there will only be ONE user.

To fully install this release, the following must occur:

1. Install the GRIS segment.

2.Those sites converting from a WWMCCS RIS need to furnish a dump of their WWMCCS databases (track db and mission db) in the ascii WWMCCS dump format.

3. Run PostInstall to move the databases to the global data area and install the AMP cron job. The cron job can be found under /h/data/global/gris/amp, called "amp_cron". It must be installed manually. Determine who it makes sense to own this cron and install it as that user.

Use the command "crontab /h/data/global/gris/amp/amp_cron", PROVIDED that the user does not have any other cronjobs. Check first with the "crontab -l" command. If cronjobs exist, then edit the amp_cron into the cron table manually.

4. Run h/GRIS/Scripts/load_data on the WWMCCS database dumps mentioned in step 3. If it was not necessary to do step 3, skip this step.

5. You will need the AMHS_CLT client to be installed on the GRIS client.

6. GRIS is configured for the following userids: GCCS. If any other userid is used, the file /h/GRIS/data/MASTER can be modified. For any problems or questions, contact the GRIS Maintenace POC. Don Ross Tel: (703)883-8777 Fax: (703) 883-8788

2.2 Setting Up a User

Use the following procedures for each user who requires access to GRIS.

Step 1: Give the user a UNIX account

Step 2: Give the user an Oracle account with the same userid as their UNIX userid. Also, make sure that the password is defined as "identified externally." (Note : this is how most GCCS user accounts are set-up).

Step 3: Log on to the database server as root and run the following script, substituting the user's userid for *userid…*

```
/h/GRISDB/Scripts/gris_enable_user userid
```

This script will check to make sure the user has an Oracle account and will then grant the GRIS_ROLE Oracle role to that user.

Step 4: Modify the `/h/GRIS/data/MASTER` file on the machine that the user will access GRIS from. Add the following line to the file, substituting the user's userid (in all caps) for *USERID…*

```
USERID    => $GRIS_LOCDAT/
```

This will specify where the user's `LOG_INIT` file will be located. In this case, we are specifying that the `LOG_INIT` filw will be located in `/h/data/local/gris`.

Step 5: Create a file called `/h/data/global/gris/values/config.USERID`, substituting the user's userid (in all caps) for USERID. In this file you can override any of the normal config options (see section 3.3) that will apply to that user. One important value that will apply to many GRIS users is the READ_ONLY value. In order to grant a GRIS user the ability to modify the database, you must set this config value to FALSE. To do so, add the following line to the user's config file…

```
READ_ONLY                              => FALSE
```

Any other config values would be specified on subsequent lines

2.3 Converting the "Old" flat-file database to Oracle

The GRIS segment must be installed under GCCS v2.1 @ or later @. As part of the install process, a cron job will begin running to pick up incoming messages. This cron job should be manually installed by the GRIS user. The cron file is called "amp_cron" and can be found in /h/data/global/gris/amp. The commands "man crontab" and "man cron" will give unix help pages on managing the cron table.

The segment includes programs for each site to convert their existing Ada databases into Oracle tables.

There are two scripts that have been set up to help conver the flat-file (pre 2.3) versions of the GRIS database to Oracle. These scripts call a couple of executable procedures that do the actual conversion. The scripts are located in the GRISDB segment and are called…

```
/h/GRISDB/Scripts/convert_codes_to_oracle
/h/GRISDB/Scripts/convert_databases_to_oracle
```

Before importing the flat-file data, you may wish to delete all the data currently residing in the GRIS Oracle database. To do so, run the following script…

```
/h/GRISDB/Scripts/erase_all_oracle_data
```

To convert the flat-file database to Oracle follow these procedures…

Step 1: Execute the following script…

```
/h/GRISDB/Scripts/convert_codes_to_oracle
```

Step 2: It may or may not prompt you for the GRIS Oracle password

Step 3: When prompted…

```
Break on error?  (y)
```

Answer yes (y) if you want the conversion program to pause and ask for you to hit <return> every time it encounters an erro. Answer no (n) if you want it to just print the error message and continue with the conversion.

Step 4: When prompted

```
Prompt for each table?  (y)
```

Answer yes (y) if you want the conversion program to ask you if you want to convert each table as it comes to it. Answer no (n) if you want it ot automaticaly convert each table.

You should get output similar to this…

```
Addressee table converted:              58 records written;     0 in error.
Aircraft_Type table converted:           9 records written;     0 in error.
Country table converted:                41 records written;     0 in error.
Event table converted:                  25 records written;     0 in error.
Frequency table converted:              23 records written;     0 in error.
Geographic_Area table converted:       125 records written;     0 in error.
Icao table converted:                    9 records written;     0 in error.
Program table converted:               216 records written;     0 in error.
Purpose table converted:                23 records written;     0 in error.
Reason_Cause table converted:            0 records written;     0 in error.
```

Step 5 : Execute the script…

```
/h/GRISDB/Scripts/convert_databases_to_oracle
```

Step 6: It may or may not prompt you for the GRIS Oracle password

Step 7: When prompted…

```
Break on error?  (y)
```

Answer yes (y) if you want the conversion program to pause and ask for you to hit <return> every time it encounters an error. Answer no (n) if you want it to just print the error message and continue with the conversion.

Step 8: When prompted

```
Prompt for each table?  (y)
```

Answer yes (y) if you want the conversion program to ask you if you want to convert each table as it comes to it. Answer no (n) if you want it ot automaticaly convert each table.

You should get output similar to this…

```
Addressee table converted:              6 records written;     0 in error.
```

```
Incoming Message table converted:      644 records written;     0 in error.
Message Lock table converted:            1 record written;      0 in error.
Mission table converted:              2268 records written;     0 in error.
Nickname table converted:              186 records written;     0 in error.
Outgoing Message table converted:        6 records written;     0 in error.
Sectional Message table converted:      37 records written;     0 in error.
Status Log table converted:            259 records written;     0 in error.
Text table converted:                   15 records written;     0 in error.
Track table converted:                 109 records written;     0 in error.
```

2.4. Setting GRIS Defaults

If the defaults are not satisfactory, have the site GRIS user set initial data values in the /h/GRIS/data/global/values/config file to the appropriate values.

To change these parameters, it will be necessary to edit the config file using a unix editor such as vi. For a detailed description of every configurable element in this file, see section 3.3. The most commonly site-unique configurable items in this file are:

LOCAL_TIME_ZONE

The numeric value that must be added to get the local time. For time zones west of Zulu, the value will be negative; for time zones east of Zulu, the value will be positive.

LOCKABLE_TRANSACTIONS

A list of booleans indicating which transaction types are to be locked out by message lock. The transaction types are the following:

Nil
Add
Change
Delete
Request
Schedule
Cancel
Delayed
Indefinite Delay
Reschedule
Premission
Execution
Launch
Relaunch
Recovery
Launch Change
Recovery Change

MONTHS_BEFORE_PURGE

The latest month counting backwards from the present month for which data is to be purged. The value should be negative. For example, a value of -3 will purge all data up to and including three months ago.

VALID_AGENCIES

A list of booleans indicating the agencies that are valid for a specific site. The agencies are the following:

Nil
PACOM

ACOM
CENTCOM
EUCOM
SOUTHCOM
ADCOM
REDCOM
SAC

DEFAULT_COMMANDS
Determines which command buttons on the Mission Query Menu will be preselected. The boolean values represent the following commands:

ACOM
CENTCOM
EUCOM
PACOM
SOUTHCOM

ROUTING_INDICATOR
The originating station routing indicator.

MESSAGE_CENTER_ID
The routing indicator for the outgoing message/switching center destination.

DEFAULT_PRINTER
The name of the default printer.

LANDSCAPE_MINUS_MARGIN
The difference between the number of lines in landscape mode with and without margins.

LANDSCAPE_PAGE_LENGTH
The default number of lines in landscape mode.

MARGIN_WIDTHS
The default margin widths. For example, UNIX puts on 5 spaces at the top and 5 spaces at the bottom, so this value would be 10.

PORTRAIT_MINUS_MARGIN
The difference between the number of lines in portrait mode with and without margins.

PORTRAIT_PAGE_LENGTH
The default number of lines in portrait mode.

In order for the Track Dictionary printout to be properly formatted, the following numeric values will have to be adjusted:

EVENT_HEADER_LINES
FOOTER_LINES
LETTERHEAD_LINES
LINES_PER_PAGE
NICKNAME_REPORT_HEADER_LINES
PAGE_WIDTH
TABLE_OF_CONTENTS_COLUMNS
TABLE_OF_CONTENTS_HEADER_LINES
TITLE_LINES

TRACK_HEADER_LINES

2.5. Converting and Installing Site Data

Those sites converting from a WWMCCS RIS need to furnish a dump of their WWMCCS databases (track database and mission database) in the ASCII WWMCCS dump format. Coordinate with the WWMCCS systems administrator. See section 3.1 for detailed instructions.

2.6. Loading Site Data

Run load_data on the WWMCCS database dumps mentioned in step 3. Depending on the size of the dumps, it may be prudent to break a very large file into two smaller files. See section 3.1 for detailed instructions.

2.7. Setting Up Codes Tables

When GRIS is installed, the code tables will have to be loaded. Most of the data in the code tables is valid across installations. A code table is a file consisting of one or more columns listing valid values for relatively static items within the system. The following code tables with the following columns are used:

Table	Columns
Addressee	Abbreviation
	Name
Aircraft Type	Mission Design Series
	Service Code
Country	Country Code
	Abbreviation
	Descriptive Name
Event	Event Code
	Abbreviated Description
	Description
Frequency	Radio Frequency
	Geographic Area
	Abbreviation
	Geographic Area Code
	Description
ICAO	ICAO Code
	Descriptive Name
	Geographic Coordinates
	Country Code
Program	Long Nickname
	Medium Nickname
	Short Nickname
	Operational_Plan
Purpose	Purpose Code
	Description
Reason-Cause	Reason Cause Code
	Description

There are also two databases which will have to be set up differently for each installation. These are:

Message Header

Text

The Addressee database consists of addressees for the following outgoing messages:

Consolidated SRO Schedule
SRO Summary
Daily Schedule
Statistical Summary

A to- or info-addressee added to the Message Header database must already exist in the Addressee code table.

The Text database consists of "boiler plate" text used in various reports and outgoing messages. This text needs to be entered only once and will automatically appear in its designated location when the report or message it belongs in is generated. The following narrative text must be entered:

Mission Recap Title
The Monthly Mission Recap Report title page.

Track Dictionary New Title
The title page of a newly generated Track Dictionary.

Track Dictionary Change Title
The title page of a set of Track Dictionary change pages.

Track Dictionary Distribution
The Track Dictionary distribution page.

Track Dictionary Remarks
The Track Dictionary remarks page.

Consolidated SRO Schedule (Part 1) Header
Text at the beginning of Part 1 of a Consolidated SRO Schedule message.

Consolidated SRO Schedule (Part 2) Header
Text at the beginning of Part 2 of a Consolidated SRO Schedule message.

Consolidated SRO Schedule (Part 2) Insert
Text inserted between the mission data and the frequency table in Part 2 of a Consolidated SRO Schedule message.

Consolidated SRO Schedule (Part 2) Footer
Text at the end of Part 2 of a Consolidated SRO Schedule message.

SRO Summary Header
Text at the beginning of an SRO Summary Message.

Statistical Summary Header
Text at the beginning of a Statistical Summary Message.

Daily Schedule Header
Text at the beginning of a Daily Schedule Message.

Daily Schedule Footer
Text at the end of a Daily Schedule Message.

Mission Schedule Header
: Text at the beginning of a Mission Schedule Message down to and including the Reason line. The DTG will be overwritten with the current date.

Mission Schedule Footer
: Text at the end of a Mission Schedule Message starting with Part/2//.

Mission Cancellation Header
: Text at the beginning of a Mission Cancellation Message down to and including the Reason line. The DTG will be overwritten with the current date.

Mission Cancellation Footer
: Text at the end of a Mission Cancellation Message starting with Part/2//.

Any narrative text for which a date is to be embedded must contain one of the following placeholders:
: DDHHMMZ MON YY

2.8. Setting AMHS For Incoming Messages

Configure the site's AMHS to profile only RECON messages to GRIS. The boolean expression "REPORTID AND RECON" will capture those messages with the word RECON on the REPORTID line.

At install time, the segment install scripts will prompt the installer to enter the directory location for the AMHS to send incoming messages to. Make sure that this location in GRIS agrees with the location in the GCCS AMHS.

2.9. Setting AMHS For Outgoing Messages

Outgoing messages are sent via the GCCS AMHS outgoing APIs. Specifically, "release_message" is called to send GRIS outgoing messages to the GCCS AMHS. Therefore, the AMHS Client segment is required by GRIS.

3.0 Installation and Configuration Support

The following sections contain detailed information for configuring GRIS under GCCS.

3.1 Loading the Databases

The loader loads the Nickname, Track, and/or Mission databases from dumps of the existing Track and Mission databases. The Track and Mission dumps are used together to load the Nickname database. The following steps explain how to setup and execute the loader.

1.) Run GRIS. Enter Database Maintenance under System. Highlight the databases you want to load and click on Initialize. This will remove all their data.

2.) If you have selected Nickname as one of the databases to load, go into your data directory and type "vi codes/program". Delete all lines of this file EXCEPT the first line. Change the number at the top to 0, RIGHT JUSTIFIED (i.e., the 0 should be in the ones place of the previous number). You can do this by typing "R" to go into replace mode and spacing over all digits of the number except the last, and replacing the last digit with 0.

3.) Run the loader by typing "run_load_data". The loader will display the following prompt:

Enter track file for nicknames. (CR to skip)

If you wish to load the Nickname database, enter the name of the Track database dump, otherwise, hit return. If you enter a file name, you will receive the following prompt:

Enter mission file for nicknames. (CR to skip)

If you wish to continue, enter the name of the Mission database dump. Otherwise, hit return. The program will continue to loop until you exit by hitting return.

4) After you exit the nickname portion of the loader, you will receive the following prompt:

Enter track file. (CR to skip)

If you wish to load the Track database, enter the name of the Track database dump. Otherwise, hit return. The program will continue to loop until you exit by hitting return.

5) After you exit the track portion of the loader, you will receive the following prompt:

Enter mission file. (CR to skip)

If you wish to load the Mission database, enter the name of the Mission database dump. Otherwise, hit return. The program will continue to loop until you exit by hitting return.

6) After you exit the mission portion of the loader, you will receive the following prompt:

CR to quit.

If you hit return, you will exit the loader. Entering any other value (which must be followed by a return) will cause the loader to loop again from the top (i.e., nicknames).

7) If at any point a nonexistent file name is entered, you will receive the following error message:

File does not exist.

You will be given another chance to enter the correct file name, or to exit if you hit return.

8) All lines in error, and lines following until the next track or mission record, will be written to a file with the same name as the Track or Mission database dump file, with an extension of _ERR. If this file is empty at the end of a run, there were no errors.

9) In testing the loader, there appears to be a bug in the runtime system's handling of heap space. The problem is that when any iterative operation is run for a long time, a Storage_Error (normally caused by either running out of heap or stack space or by computing a nonexistent address) is raised.

Tests show nearly all allocated memory is being returned; thus, the runtime system is suspect. Tests revealed that when running the loader - loading the Mission database - a Storage_Error was raised after loading about 6 months worth of data.

To avoid this, simply split your Mission database dump file in half (at the beginning of a new mission, of course), run the loader once on the first half, EXIT THE LOADER COMPLETELY, and then rerun the loader on the second half.

3.2 Backing up and Restoring the GRIS part of the Oracle Database

To backup the GRIS Oracle database, do the following steps…

Step 1 : Execute the following script…

```
/h/GRISDB/Scripts/create_database_dump dump_file
```

Where dump_file is the file to write the data to.

The script will use Oracles exp command to create a dump file that can later be re-imported. You can save this dump file on disk or tar it off to tape.

To restore the GRIS database from one of these backups, do the following…

Step 1 : Execute the following script…

```
/h/GRISDB/Scripts/import_database_dump dump_file
```

Where dump_file is the dump file containing the data.

3.3 Master File

The MASTER file tells the program being executed the locations of the config and message files for the project and the directory where the current user's LOG_INIT file is located. The location of UI setup files is also returned but is no longer used.

The MASTER file is comprised of three areas, labeled, PROJECTS, USERS, and TRIGGERS.

The PROJECTS area contains the following values for each project:

<project-name>_CONFIG - The path and name of the config file.
<project-name>_MESSAGE - The path and name of the message file.

<project-name>_UI - Not used at present.

The USERS area contains an entry for each user of the system, followed by the location of the LOG_INIT file ($GRIS_SEGDAT). The location must end in a separator (e.g., '$GRIS_SEGDAT/'). If no location is specified, it will be assumed that the LOG_INIT file is in the current directory.

The TRIGGERS area is comprised of two entries:

PROJECT
USER

If nothing follows these entries, the user will be prompted for the project name and the user id. If values are specified for these entries, the user will not be prompted, and the system will search the MASTER file for them. If a match is not found (whether the project name and user id are entered by the user or contained in the TRIGGERS area of the MASTER file), the error message, "Master Config File not in proper format" will be displayed.

The location of the MASTER file is specified by the $MASTER environment variable. This value must end in a separator (e.g., '/', '.', etc). If the MASTER file cannot be found or cannot be opened for some reason, the error message "Config File <$MASTER>MASTER could not be opened" will be displayed.

3.4 Config File

GRIS' config file contains all of the tunable parameters and their values. Users can edit any of the elements in the config file to suit their site's needs. The config file gets read and interpreted during system startup. Therefore, any changes to the file will not take effect until the next execution.

The following is a complete listing of the configurable parameters:

3.4.1 Archive

ARCHIVE_FILE_PATH
: The path to the archive data files. The mission archive files are stored in subdirectories with the name of the year of the mission and files with the name of the month and year of the mission, the resulting format being YYYY/MMMYYYY. The track archive file is defined by TRACK_ARCHIVE_FILE.

TRACK_ARCHIVE_FILE
: The name of the track archive file. It does not include a path.

3.4.2 Automated Message Processor

AMP_DEBUGGING
: Turns on/off printing of the message format table.

AMP_FAILED_FLAG_NAME
: A flag file indicating that AMP has failed.

AMP_LOG_FILE
: A log file consisting of a time stamp, the date of the message, and the message state for each message processed.

INCOMING_MESSAGE_FILE_NAME
: The file AMP processes.

MESSAGE_FORMAT
: 5157 or USMTF format.

MAX_ERRORS_IN_SEQUENCE
: The number of back-to-back errors AMP will allow before terminating with an error status.

MAX_WARNINGS
: The number of times an AMP warning (e.g. that a high precedence message has been received, or that AMP has failed) will be flashed before turning off.

MAXIMUM_MESSAGE_FORMAT_TABLE_LINE_LENGTH
: The maximum length of a line in the message format table.

MAXIMUM_MESSAGE_TOKEN_IMAGE_LENGTH
: The maximum length of a message token. A message token is any sequence of characters in a message between two delimiters (e.g., /)

MAXIMUM_MISSION_HOURS
: The maximum allowable number of hours between departure and arrival times.

MAXIMUM_MISSION_MINUTES
: The maximum allowable number of minutes between scheduled and actual departure/arrival times.

MESSAGE_DUPLICATE_SAVE
: Whether duplicate messages are to be saved in the Message database or thrown away.

MESSAGE_FORMAT_TABLE_NAME
: The name of the message format table.

MESSAGE_HEADER_TABLE_DELIMITER
: The delimiter indicating section breaks in the message header table.

MESSAGE_HEADER_TABLE_MAX_LINE_LENGTH
: The maximum line length in the message header table.

MESSAGE_HEADER_TABLE_NAME
: The name of the message header table.

PRECEDENCE_FLAG_NAME
: A flag file indicating that a high precedence message has been received.

RECON_ERROR_NAME
: The file by which error messages are passed from the recon table navigator to the application.

RECON_MESSAGE_NAME
: The file created by the the recon message builder.

RECON_PROMPT_NAME
: The file by which choice lists are passed from the recon table navigator to the application.

3.4.3 Break

BREAK_FILE
: The file defining what processes will cause the break window to be displayed. The lines in this file are of the form <subsystem>. <operation>.

BREAK_FLAG
: The name of the flag file created by the break driver telling GRIS to cancel the current operation. The user id will be appended to the file name to make it unique for the user. In addition, the name is used as a base for GRIS to tell the break driver to pop up the break window, to pop down the break window, or to quit. In this case the suffixes _up, _down, _quit are appended to the name and before the user extension.

BREAK_WAIT
: The number of seconds the break driver will wait for checking for BREAK_FLAG from GRIS.

3.4.4 Codes

The following code tables all have the same configurable values:

Addressee
Aircraft Type
Country
Event
Frequency

Geographic Area
ICAO
Program
Purpose
Reason-Cause

FILE_NAME
: The name of the code table.

MAXIMUM_ROWS
: The maximum number of rows allowed in the code table.

SCHEMA
: The widths of the code table columns.

SORTING_COLUMN
: The code table column on which sorting will be done by default.

UNIQUENESS
: Each code table contains one or more columns. This value determines whether the items in the columns are to be unique. This is a boolean value, one for each column in the code table. The code tables have the following columns:

Addressee	Abbreviation
	Name
Aircraft Type	Mission Design Series
	Service Code
Country	Country Code
	Abbreviation
	Descriptive Name
Event	Event Code
	Abbreviated Description
	Description
Frequency	Radio Frequency
Geographic Area	Abbreviation
	Geographic Area Code
	Description
ICAO	ICAO Code
	Descriptive Name
	Geographic Coordinates
	Country Code
Program	Long Nickname
	Medium Nickname
	Short Nickname
	Operational_Plan
Purpose	Purpose Code
	Description
Reason-Cause	Reason Cause Code
	Description

3.4.5 Databases

DATABASE_DEBUGGING

Turns on/off trace messages having the LOG_INIT trigger of SEARCH.

DATABASE_OWNER_NAME

The owner of the Oracle tables. This is needed by other users to prepend to the table names.

DATABASE_USER_NAME

The logon value for Oracle.

DATABASE_USERS

The maximum number of users the database must support.

DATABASE_WAIT

The number of seconds a process will wait for a shared resource that is currently reserved by another user before timing out.

JOINS_ON

Controls whether database table joins will be done. When on, sorting by Geographic Area, ICAO, Program, and Purpose codes will be by SORTING_COLUMN. However, it may be turned off if database table space becomes a problem.

MAX_DATABASE_GETS

The number of items that will be retrieved both initially and after any selection of the More button on an index display.

MAX_TIME_BETWEEN_SNAPSHOTS

The number of minutes between database snapshots. Database snapshots force a disk write so as to preserve data in case the system goes down.

REMARK_TABLE_NAME

The name of the remark table, which stores remarks belonging to database items.

SQL_QUERY_LENGTH
: The maximum length of a SQL query string.

SUMMARY_TABLE_NAME
: The name of the summary table, which stores index screen images of database items.

The following databases all have the same configurable values:

Addressee
Incoming Message
Message Lock
Mission
Nickname
Outgoing Message
Sectional Message
Status Log
Text
Track

DATABASE_NAME
: The name of the database.

DATABASE_SIZE
: The maximum size of the database.

KEYS_UNIQUE
: If true, then each item in the database must contain at least one Unique key. The databases have the following keys:

Addressee	Outgoing Message
Incoming Message	Message Key Entry (Date Time Group
	Addressee
	Version Number)
	Error Code
	Message Type
	Transaction Type
	Period
	Primary Track Number
	Secondary Track Number
	Tertiary Track Number
	Quaternary Track Number
	Quinary Track Number
	Primary Program
	Secondary Program
	Tertiary Program
Message Lock	[dummy key]
Mission	Mission Number
	Error
	Primary Area
	Secondary Area
	Tertiary Area
	Icao
	Program
	Primary Purpose
	Secondary Purpose
	Track Number

	Status Result
	Scheduling
	Period
	Takeoff Time
	Landing Time
Nickname	Nickname Key Entry (Agency Program)
	Program
	Status
Outgoing Message	Outgoing Message
Sectional Message	Date Time Group
	Addressee
	Section
Status Log	Time Stamp
	Status Log Key Entry (Entry Type
	Date Time Group
	Addressee
	Version Number
	Mission Number
	Track Number)
Text	Page
Track	Track Key Entry (Agency
	Program
	Track Number)
	Program
	Track Number
	Track Status
	Primary Area
	Secondary Area
	Tertiary Area
	Primary Purpose
	Secondary Purpose

In addition, the Addressee, Mission, and Track databases have the following special values:

ADDRESSEE_TABLE_NAME
: The name of the addressee table, containing to and info addressees.

MISSION_MESSAGE_TABLE_NAME
: The name of the mission message table, containing message header information.

MISSION_MESSAGE_REMARK_TABLE_NAME
: The name of the mission message remark table, containing special remarks.

TRACK_DICTIONARY_INFO_TABLE_NAME
: The name of the track dictionary info table, containing information about the track relevant to the track dictionary.

TRACK_EVENT_TABLE_NAME
: The name of the track event table, containing the events belonging to the track.

TRACK_ORBIT_TABLE_NAME
: The name of the track orbit table, containing the orbits belonging to the track.

3.4.6 Enumerations

These values give the names of the files in which the string representation of the following types are stored:

CLASSIFICATION

DESTINATION
ORIENTATION
USER

3.4.7 Garbage Collection

GARBAGE_COLLECTION_THRESHOLD
: The number of bytes allocated to system-wide dynamically allocated variables that will trigger a message to return to the main menu for memory reclamation.

HEAP_SIZE
: The number of bytes available to system-wide dynamically allocated variables. This memory space is reclaimed at the main menu.

3.4.8 GRIS

COMPARISONS_PER_MINUTE
: A value to help estimate the length of time a query will take.

COUNTER_FILE
: A table listing the number of unreclaimed dynamically allocated variables will be written to this file.

LOCAL_TIME_ZONE
: The numeric value that must be added to Zulu time to get the local time. For time zones west of Zulu, the value will be negative; for time zones east of Zulu, the value will be positive.

LOCKABLE_TRANSACTIONS
: A list of booleans indicating which transaction types are to be locked out by message lock. The transaction types are the following:

Nil
Add
Change
Delete
Request
Schedule
Cancel
Delayed
Indefinite Delay
Reschedule
Premission
Execution
Launch
Relaunch
Recovery
Launch Change
Recovery Change

MAILBOX_WAIT
: The number of seconds internakl mesage traffic will wait for a response before timing out.

MONTHS_BEFORE_PURGE
: The latest month counting backwards from the present month for which data is to be purged. The value should be negative. For example, a value of -3 will purge all data up to and including three months ago.

PROJECT_RELEASE
: The version number of the current release.

REFRESH_TIME
: The interval at which the MMI will query for dynamic data.

SCRATCH_AREA
: The path to a directory where temporary files can be written.

SECONDS_TO_TIMEOUT
The number of seconds to wait for a spawned job to acknowledge its completion before timing out.
SECURITY_CLASSIFICATION
The security classification of the system.
VALID_AGENCIES
A list of booleans indicating the agencies that are valid for a specific site. The agencies are the following:

Nil
PACOM
ACOM
CENTCOM
EUCOM
SOUTHCOM
UNUSED_1
UNUSED_2
UNCOM

3.4.9 Help

ERROR_HELP
The path and name of the error message help file.
HELP_FILE_PATH
Where the help files are located.
MESSAGE_HELP
The path and name of the incoming message detail help file.

3.4.10 Incoming Messages

ADDRESSEE_BUTTON_ON
Determines whether the "With Addressees" button on the Incoming Message Query Menu will be preselected.
DEFAULT_MESSAGE_TYPES
Determines which message type buttons on the Incoming Message Query Menu will be preselected. The boolean values represent the following message types:

Recon_1
Recon_2
Recon_3
Recon_4

DEFAULT_TRANSACTION_TYPES
Determines which transaction type buttons on the Incoming Message Query Menu will be preselected. The boolean values represent the following transaction types:

Add
Cancel
Change
Definite_Delay
Delete
Execution
Indefinite_Delay
Launch
Premission
Recovery

Relaunch
Request
Reschedule
Schedule

DIAGNOSTIC_BUTTON_ON
: Determines whether the "With Diagnostics" button on the Incoming Message Query Menu will be preselected.

KEEP_LATEST_MESSAGE_ONLY
: If true, discards previous revisions of a message; if false, keeps them.

MESSAGE_ERROR_BUTTON_ON
: Determines whether the "With Errors" button on the Incoming Message Query Menu will be preselected.

3.4.11 Lists

All lists have the following configurable values:

NAME
: A five-character mnemonic uniquely identifying the list.

SIZE
: The size of the list.

3.4.12 Maps

EXPORT_MAP_FILE
: The name of the file to which map data will be written.

MAP_EXECUTABLE
: The name of the mapping executable.

3.4.13 Mission Colors

The following colors should be set with a Motif color name. They are used for the bar charts in the Chart/Matrix displays.

ABORT_INCOMPLETE_COLOR
CANCELED_COLOR
COMPLETED_COLOR
DELAYED_COLOR
DELETED_COLOR
FLYING_COLOR
INDEFINITELY_DELAYED_COLOR
REQUESTED_COLOR
SCHEDULED_COLOR

3.4.14 Mission Reports

DAILY_MISSION_LOG_REPORT
: The name of the file to which the Daily Mission Log is to be written.

DEFAULT_HOUR
: The hour at which the Daily Mission Log is to start and stop. Used with DEFAULT_MINUTE.

DEFAULT_MINUTE
: The minute at which the Daily Mission Log is to start and stop. Used with DEFAULT_HOUR.

GRAPH_START
: Where the bar chart portion of the Mission Chart/Matrix displays starts. This must follow the MISSION_SORT_KEY_FIELD_WIDTH.

GRAPH_STOP
: Where the bar chart portion of the Mission Chart/Matrix displays stops.

MAX_LINE_FEEDS
: The maximum number of lines to be shown in the Recap display.

MISSION_CHART_MATRIX_REPORT
: The name of the file to which the Mission Chart/Matrix Reports are to be written.

MISSION_OVERVIEW_REPORT
: The name of the file to which the Mission Overview report is to be written.

MISSION_SORT_KEY_FIELD_WIDTH
: The width of the sort keys column in the MissionChart/Matrix and Overview reports.

MONTHLY_MISSION_RECAP_REPORT
: The name of the file to which the Mission Recap Monthly Report is to be written.

3.4.15 Missions

DEFAULT_COMMANDS
: Determines which command buttons on the Mission Query Menu will be preselected. The boolean values represent the following commands:

ACOM
CENTCOM
EUCOM
PACOM
SOUTHCOM

DEFAULT_STATUSES_AND_RESULTS
: Determines which status-result buttons on the Mission Query Menu will be preselected. The boolean values represent the following status-results:

Available
Canceled
Delayed
Deleted
Flying As Scheduled
Flying Deviation
Flown Abort Complete
Flown Abort Incomplete
Flown As Scheduled
Flown Deviation
Indefinite Delay
Requested
Rescheduled
Scheduled

DEFAULT_TIMES
: Determines which times on the Mission Query Menu will be preselected. The boolean values represent the following times:

Today
Tomorrow
Yesterday
Current Month
Next Month
Last Month

Current 4 Days (Today plus next three days)

LANDING_TIME_GRACE_PERIOD
The number of minutes a mission must be late in landing before it is declared to be overdue.
MISSION_ERROR_BUTTON_ON
Determines whether the "With Errors" button on the Mission Query Menu will be preselected.
TAKEOFF_TIME_GRACE_PERIOD
The number of minutes a mission must be late in taking off before it is declared to be overdue.
SCHEDULING_BUTTON_ON
Determines whether the "QRC/Contingency" button on the Mission Query Menu will be preselected.
STATUS_RESULT_FILE
The name of the file defining valid mission status-result transitions.

3.4.16 Monitors

MISSION_MONITOR
The name of the mission monitor invocation script.
MONITOR_INTERVAL
The number of seconds between calculations of the percentage of an operation that has been completed. This value is displayed in the status window if the MONITOR trigger is turned on.
MONITOR_HOURS_BEFORE_CURRENT
All missions whose takeoff times are between the specified number of hours before the present time and MONITOR_HOURS_AFTER_CURRENT will be displayed by the Mission Index Monitor.
MONITOR_HOURS_AFTER_CURRENT
All missions whose takeoff times are between MONITOR_HOURS_BEFORE_CURRENT and the specified number of hours after the present time will be displayed by the Mission Index Monitor.
STATUS_LOG_MONITOR
The name of the status log monitor invocation script.

3.4.17 Outgoing Messages

CONSOLIDATED_SRO_SCHEDULE_DECLASSIFICATION
The declassification instruction for the Consolidated SRO Schedule Message.
CONSOLIDATED_SRO_SCHEDULE_MESSAGE_FILE
The name of the file to which the Consolidated SRO Schedule Message will be written.
DAILY_SCHEDULE_DECLASSIFICATION
The declassification instruction for the Daily Schedule Message.
DAILY_SCHEDULE_MESSAGE_FILE
The name of the file to which the Daily Schedule Message will be written.
INCOMING_MESSAGE_FILE
The name of the file to which a retransmitted incoming message will be written.
MAX_OUTGOING_MESSAGE_LENGTH
The maximum allowable length of an outgoing message.
MESSAGE_TOMORROW_BUTTON_ON
Determines whether the "Tomorrow's" button on the Daily Schedule Message Generation Menu will be preselected.
MISSION_CANCELLATION_MESSAGE_FILE
The name of the file to which the Mission Cancellation Message will be written.
MISSION_SCHEDULE_MESSAGE_FILE
The name of the file to which the Mission Schedule Message will be written.
ROUTING_INDICATOR
The code of the installation from which an outgoing message is transmitted.
ROUTING_LINE_FORM
The format of Format Line 2, where lower case letters are placeholders.They are the following:

p - precedence
c - classification
r - routing indicator
j - julian date
t - hour and minute

SERIAL_NUMBER_FILE
A file containing the current serial number for outgoing messages.

SRO_SUMMARY_DECLASSIFICATION
The declassification instruction for the SRO Summary Message.

SRO_SUMMARY_MESSAGE_FILE
The name of the file to which the SRO Summary Message will be written.

STATISTICAL_SUMMARY_DECLASSIFICATION
The declassification instruction for the Statistical Summary Message.

STATISTICAL_SUMMARY_MESSAGE_FILE
The name of the file to which the Statistical Summary Message will be written.

TRANSMIT_EXECUTABLE
The name of the incoming/outgoing message (re)transmit executable.

3.4.18 Password

MINIMUM_PASSWORD
The minimum allowable password length.

PASSWORD_FILE_NAME
The name of the file to which the encrypted password is stored.

PASSWORD_LEVELS
The number of password levels maintained by the system.

3.4.19 Pointers

All sharable components have the following configurable value:

NAME

3.4.20 Printing

DEFAULT_DESTINATION
The Print Control Menu can send its output to either a file or a printer. This value specifies which is to be the default. The following values are allowed:

FILE
PRINTER

DEFAULT_ORIENTATION
Files can be printed in either portrait or landscape orientation. This value specifies which is to be the default. The following values are allowed:

PORTRAIT
LANDSCAPE

DEFAULT_PRINTER
The name of the default printer.

LANDSCAPE_MINUS_MARGIN
The difference between the number of lines in landscape mode with and without margins.

LANDSCAPE_PAGE_LENGTH
The default number of lines in landscape mode.

MARGIN_WIDTHS
: The default margin widths. For example, UNIX puts on 5 at the top and 5 at the bottom, so this value would be 10.

PORTRAIT_MINUS_MARGIN
: The difference between the number of lines in portrait mode with and without margins.

PORTRAIT_PAGE_LENGTH
: The default number of lines in portrait mode.

PRINT_FILE_NAME
: The name of the file to which the print output is to be written.

PRINT_EXECUTABLE
: The name of the print executable.

PRINT_RESPONSE_FILE_NAME
: The name of the file to which the print job number is to be written.

3.4.21 Strings

These values give the names of the files in which the legal characters of the following types are defined. These files must contain a list of ASCII characters followed by a boolean value indicating whether or not they are legal for the type.

FILE_NAME
FREE_TEXT
PASSWORD
PRINTER_NAME

3.4.22 Tracks

DEFAULT_TRACK_SORT_ORDER
: Determines which sort order on the Track Query Menu will be preselected. The following values are allowed:

NONE
ACTIVE
REQUESTED
DELETED
INACTIVE

DEFAULT_TRACK_STATUS
: Determines which track status on the Track Query Menu will be preselected. The following values are allowed:

BY_AGENCY
BY_PROGRAM
BY_TRACK

3.4.23 Track Dictionary

In order for the Track Dictionary to be properly formatted, the following numeric values will have to be adjusted:

EVENT_HEADER_LINES
FOOTER_LINES
LETTERHEAD_LINES
LINES_PER_PAGE
NICKNAME_REPORT_HEADER_LINES
PAGE_WIDTH

TABLE_OF_CONTENTS_COLUMNS
TABLE_OF_CONTENTS_HEADER_LINES
TITLE_LINES
TRACK_HEADER_LINES

3.4.24 Track Reports

TRACK_DICTIONARY
: The name of the file to which the Track Dictionary is to be written.

TRACK_REPORT
: The name of the file to which the Track Reports are to be written.

3.4.25 User

These are values that typically are user-specific, and so would be put in the user config file. However, in the event that they are not there, or that a config file does not exist for that user, the default values should be included here.

CAN_TRANSMIT
: User has transmit privileges.

CONFIRMATION_ON
: If true, will pop up confirmation screens when an operation has been successfully completed; if false, the confirmation screens are suppressed.

PROCESS_ON_SAVE
: Controls whether reprocessing of a message is a one - or two- step process: a value of true causes reprocessing to be automatically invoked after saving a message; a value of false means that the user must click on the Save button to save the message and on the Process button to reprocess it. Whether this value is set to true or false depends on whether you want to reprocess a message every time you save it.

READ_ONLY
: User does not have write privileges.

REFRESH_QUERIES
: If true, query values will be removed upon returning from an operation to the query menu; if false, they will still be there.

3.5 Configuring the Automated Message Processor

GRIS has it's own Automated Message Processor (AMP) for parsing RECON messages. AMP is designed to be partially configurable by changing values in the following tables:

Message Header Table
Message Format Table
Status-Result Table

3.5.1 Message Header Table

The Message Header Table is a state transition table defining the processing of a Recon message header. It allows this procesing to be configurable. It contains three sections separated by MESSAGE_HEADER_TABLE_DELIMITER, defined in the config file.

Section 1 associates the internal names of the state transition table, Zero through Seven, with the actual values they represent. A given internal name may represent more than one value.

Section 2 is the state transition table itself. The initial state is Zero. Read down to the current state and across to the next token encountered. At that position will be the new state. For example, if the current state is Zero and a Precedence_Token is encountered, the new state will be One. Each position in the state transition table must have Zero through Seven or one of the following values:

Fatal
: A fatal error has occurred. Stop processing.

Non_Fatal
: A non-fatal error has occurred. Continue processing.

Final
: The final state has been reached without errors.

Non_Fatal_Final
: A non-fatal error has occurred. Stop processing.

Section 3 contains the following keywords:

Final
Fatal

This causes the procedure processing the message header to exit, either normally (if Final is encountered) or with an error (if Fatal is encountered).

3.5.2 Message_Format Table

The Message Format Table allows limited configurability of AMP message processing. It is comprised of 4 sections, one for each Recon message type. Each section contains the following columns:

REPEAT
SET_OCCURRENCE
SET_ID
FIELD_OCCURRENCE
CONDITIONAL
FIELD_ID
VALUE

The REPEAT column defines the starting and stopping points of loops. It uses the key words BEGIN and END. When AMP reaches an END, it tries to go back to the corresponding BEGIN. If it finds the field corresponding to the BEGIN, it starts another iteration of the loop; if not, it exits the loop and goes on. BEGINs and ENDs may be nested. The corresponding BEGINs and ENDs are those at the same level of nesting. If multiple BEGINs and/or ENDs occur on the sameline, they must be enclosed in parentheses and separated by commas, for example, (BEGIN,END) or (END,END).

The SET_OCCURRENCE column must contain one of the following entries:

MANDATORY
CONDITIONAL
OPTIONAL

If a set is MANDATORY, it must appear in the message; if it is OPTIONAL it may or may not appear in the message. If it is CONDITIONAL, then the condition defining its validity must appear in the CONDITIONAL column, discussed below.

The SET_ID column contains the name of the set (e.g., REPORTID).

The FIELD_OCCURRENCE column has the same rules as the SET_OCCURRENCE column, except that instead of defining the modality of a set, it defines the modality of a field within a set.

The CONDITIONAL column defines the condition for a CONDITIONAL set or field. An entry in this column must contain four values, separated by commas:

- The condition, which must be M (for mandatory) or P (for prohibited)
- The set name from the SET_ID column
- The field name from the FIELD_ID column
- The value of the field for which the condition is either mandatory or prohibited. For example, if a set or field is prohibited if the TRANSACTION_TYPE field within the REPORTID set is ADD, this column would read P,REPORTID,TRANSACTION_TYPE,ADD. If multiple field values define the condition, they must be enclosed in parentheses and separated by commas, for example P,REPORTID,TRANSACTION_TYPE,(ADD,CHANGE).

The FIELD_ID column contains the name of the field (e.g., RECON_NUMBER).

The VALUE column contains the values of all FIELD_MARKs and SET_ENDs as well as any nonstandard field values (e.g., alternate spellings or misspellings). If there are multiple values that belong in this column, they must be enclosed in parentheses and separated by commas, for example, (REPORT,REPROT).

3.5.3 Status_Result Table

The Status_Result Table contains three columns:

From	The current Status_Result
Transaction	The requested state transition
To	Any legal resulting Status_Result (there may be more than one)

If a message contains a state transition request that is not defined in the Status_Result table, then that line will be flagged as an error.

4.0 Scripts

GRIS has several external interfaces which interact with the environment. These are in the form of script files, which can be tailored to meet the needs of the environment at a given site. All scripts are located at /h/GRIS/Scripts.

The following is a list of the script files that GRIS uses:

GRIS_warn.csh
: Used as a popup xterm during installation to warn the user that GRIS should only be installed if necessary for that site.

GRIS_info.csh
: Used as a popup xterm during installation to inform the user that the GRIS userid needs to be added to the MASTER file and to get the directory name for incoming messages.

amhs
: Displays the results of AMHS processing.

check_oracle
: Checks whether the user has access to Oracle and the JPDRPT_ROLE role..

display_map
: Displays tracks on a map of the world. (Not available in this release.)

export_map
: Writes track event locations to a file. (Not available in this release.)

out_msg
: Spawned by the Transmit button in the Incoming and Outgoing Message Detail screens. This should be interfaced to the AMHS at the site. Edits and sectionalizes (if necessary) outgoing messages before calling release_message (AMHS script) and reporting transmission results. Out_msg uses the scripts patch_trans, mparter and amhs.

pro_inc
: Spawned by the amp_cron cron job, pro_inc checks for incoming messages in a given directory, then sends them to the AMP process. Pro_inc uses the script "fixit" for editing of incoming messages. Should there be a problem, pro_inc performs a backup, and alerts the user.

rrprint
: Spawned by the print control menu. Sends a file to the specified printer.

The following scripts are used to start GRIS:

initialize
: Sets up elaboration and execution flag files.

run_application
: Calls initialize, the specified application and terminate.

terminate
: Displays an error message in case of an elaboration error.

The following scripts are used to run GRIS:

run_amp
: Runs the Automated Message Processor (AMP).

run_gris
: Runs the main application. Another run_gris script is located in the $GRIS_PROGS directory, so that GRIS can be executed from the launch button.

run_load_data
: Runs database loader/archive retrieval program.

run_mission_monitor
: Runs the Mission Monitior..

run_gris
: Runs the Status Log Monitor.

5.0 Acronyms

ACOM	U.S. Atlantic Command
AMHS	Automated Message Handling System
AMP	Automated Message Processor
AUTODIN	Automated Digital Network
CDRL	Contract Data Requirement List
CENTCOM	U.S. Central Command
COE	Common Operating Environment
CSCI	Computer Software Configuration Item
DISA	Defense Information Systems Agency
EUCOM	U.S. European Command
GCCS	Global Command and Control System
GRIS	GCCS Reconnaissance Information System
JEKE	Center For ApplicationsEngineering, Command and Control Applications Dept.
JMAS	Joint Mission Application System
JRC	Joint Reconnaissance Center
NMCC	National Military Command Center
PACOM	U.S. Pacific Command
RIPS	Reconnaissance Information Processing System
ROD	Reconnaissance Operations Division
SOUTHCOM	U.S. Southern Command
SRO	Sensitive Reconnaissance Operations
SRS	Software Requirement Specifications